Bayesian Analysis (extreme introduction)

Theme of workshop (and book): Analyzing HMs using both classical and Bayesian methods b/c if you want to be an effective user, you need to understand and be able to use both.

1. Parametric inference
2. Prior distributions.... everything is a random variable
3. Posterior inference
4. MCMC
5. Relevance to hierarchical models such as occupancy models

1. Parametric Inference

- Both Bayes or classical/frequentist/likelihood paradigms are *parametric*
  - This means: we make explicit probability model assumptions about "data" and "biological process" and **assume those assumptions are truth**
  - Not an "approximation to truth" but actual truth. (try saying this in a paper)
- With explicit probability models describing possible observables (outcomes of an experiment) we can create the **joint probability distribution** of all random variables
  - Bayes or classical inference is based on doing some kind of analysis of this joint probability distribution using numerical or computational methods.

1. Parametric Inference

Two main flavors of parametric inference

- **Classical inference**: Joint probability distribution of observations forms the **likelihood**. We maximize it to obtain MLEs and do other fun things to it.

- **Bayesian inference**: Based on the posterior distribution – proportional to the joint probability distribution of all random quantities in the model: data, random effects, parameters.

2. Random variables

How are random variables used in Bayesian and classical analysis?

• The **DATA** are realizations of a random variable, from some probability distribution:

$y \sim f(y|\theta)$, θ is a parameter.

e.g., Capture-recapture study over 5 occasions: $y \sim \text{Binomial}(5, p)$

e.g., Occupancy study over 5 occasions: $y \sim \text{Binomial}(5, p)$

• In addition, in Bayesian analysis **all unknown quantities** in a model are random variables. Includes "parameters" but also latent variables ("random effects")

2. Random variables

- The prior distribution: $\theta \sim g(\theta)$ is a *prior distribution* which describes possible values of the parameter θ in the absence of information from data. Prior to the experiment.

  Prior can be chosen to reflect prior "beliefs" or information

  Can be subjective, but this doesn't really do us much good in practice. Over-sold aspect of Bayesianism.

  Vague or non-informative (also sometimes "objective" priors): Prior might be chosen to reflect absence of specific information

- Example: Capture-recapture. Before I collect data, I don't know anything about p: $p \sim \text{Uniform}(0, 1)$.

- Ignorance: **NOT TRUE!!!!** You can't possibly initiate a study without having pretty strong ideas of what you're going to observe. Yet Bayesians all the time use "non-informative" or "flat" priors.

2. Random variables

- Frequentists have random effects too but they like to make a distinction between random effects and parameters (not random). Why?
- Bayes is coherent: *Everything* is a random variable

The Sad Caricature: “Parameters are random”

AKA: the “fixed but unknown” canard

“*The difference between Bayesian and frequentist inference is that, in Bayesian analysis, parameters are random but, in frequentist analysis, they are fixed but unknown.*”

- For a given data set, parameter values are *fixed but unknown* just like a frequentist. i.e., there is a single data-generating value, picked by nature, and you don’t know it.
- They are not “random” anyhow but rather *realizations of a random variable*. Nature generated them from some distribution. (is that the same thing? Dude, thats totally random.)
- Even qualified, “random variables” is not a diagnostic characteristic of Bayesianism: Random effects are common everywhere – they are features of models, not of inference paradigms.

How is the randomness used to facilitate inferences? That is the key issue.

3. Posterior Inference

The **posterior distribution**: The probability distribution of everything that is unknown given what is known. Usually we just say "parameters given data":

$$f(p|y)$$

Arises by use of basic rules of probability, because everything is a random variable. Bayes' rule:

$$f(p|y) = \frac{f(y|p)g(p)}{f(y)}$$

- it is a probability distribution **for the parameter(s)**!
- characterize uncertainty in the parameter values using explicit probability statements
- e.g., $Pr(L \leq p \leq U)$ = "Bayesian confidence interval" (remember in classical analysis it is the *interval that is random*).
- In general, report summaries of the posterior distribution: mean, mode, variance, etc..

3. Posterior Inference

- Posterior inference is a direct consequence of viewing things as random variables and our natural tendency to characterize uncertainty using probability.

- Sometimes the posterior is directly proportional to the likelihood in which case the inferences Bayesians and frequentists make are hardly different numerically. But conceptually they are extremely different.

- Bayesians make probability statements about unknown parameters. Frequentists do not. They make probability statements about *procedures* that they concoct. So you have things like (frequentist) "If I repeated my study a large number of times then, in the long run, 95% of the data sets generated will produce confidence intervals that contain the truth."

- But what about this one data set that I have? The Bayesian says "given the data you actually have, there is a 95% chance that N is between N_L and N_U"

Illustrations of Parametric Inference Ideas for Presence/Absence Data (Logistic Regression)

- **Observations**:

$$y_1, y_2, \ldots, y_n$$

- **Probability model for observations**

e.g., logistic regression (for binary data):

(1) Binomial probability mass function

$$y_i \sim \mathrm{Bin}(y_i; 1, p_i)$$

with

$$logit(p_i) = \beta_0 + \beta_1 x_i$$

(2) *and* $y_1, \ldots, y_n$ are mutually *independent*

- What do we do with this probability model to achieve formal inference about $\boldsymbol{\beta} = (\beta_0, \beta_1)$?

Classical Inference

Joint distribution:

$$f(y_1, y_2, \ldots, y_n; \beta_0, \beta_1) = \left\{ \prod_{i=1}^{n} \mathrm{Bin}(y_i; \boldsymbol{\beta}) \right\}$$

Likelihood = joint distribution regarded as a function of $\boldsymbol{\beta}$:

$$L(\beta_0, \beta_1; y_1, y_2, \ldots, y_n) \equiv \left\{ \prod_{i=1}^{n} \mathrm{Bin}(y_i | \boldsymbol{\beta}) \right\}$$

- Maximize it to obtain the MLE $\hat{\boldsymbol{\beta}}$
- 2nd derivative of $log(L)$ w.r.t. $\hat{\boldsymbol{\beta}}$ is the *Fisher Information* – inverse is "asymptotic variance"
- Function of the parameters (not y)!
- It is **not** a probability distribution

Example: Ordinary Logistic Regression

Binary state variable:

$$z_i \sim \text{Bin}(\psi_i)$$

$$logit(\psi_i) = \beta_0 + \beta_1 x_i$$

R script `likeBayes.R`

```
# -------------------------- Simulate data ------------------------------
# Create a covariate called vegHt
nSites <- 100
set.seed(443)                # so that we all get the same values of vegHt
vegHt <- runif(nSites, 1, 3) # uniform from 1 to 3

# Suppose that occupancy probability increases with vegHt
# The relationship is described by an intercept of -3 and
#    a slope parameter of 2 on the logit scale
# plogis is the inverse-logit (constrains us back to the [0-1] scale)
psi <- plogis(-3 + 2*vegHt)

# Now we go to 100 sites and observe presence or absence
# Actually, let's just simulate the data
z <- rbinom(nSites, 1, psi)
```

Strategy for classical estimation based on likelihood: Express the likelihood as an **R** function and then use the standard function `optim` or `nlm` to maximize it.

```
# This is the negative log-likelihood.
negLogLike <- function(beta, y, x) {
    beta0 <- beta[1]
    beta1 <- beta[2]
    psi <- plogis(beta0 + beta1*x) # inverse-logit
    likelihood <- psi^y * (1-psi)^(1-y) # same as:
#   likelihood <- dbinom(y, 1, psi)
    return(-sum(log(likelihood)))
}

# Look at (negative) log-likelihood for 2 parameter sets
negLogLike(c(0,0), y=z, x=vegHt)
negLogLike(c(-3,2), y=z, x=vegHt) # Lower is better!


# Let's minimize it
starting.values <- c(beta0=0, beta1=0)
opt.out <- optim(starting.values, negLogLike, y=z, x=vegHt, hessian=TRUE)

mles <- opt.out$par     # MLEs are pretty close to truth
```

Bayesian Inference

- Parameters are realizations of random variables:

$$g(\beta_0, \beta_1) \equiv [\beta_0, \beta_1] \qquad \text{(prior distribution)}$$

- Joint distribution:

$$f(y_1, y_2, \ldots, y_n, \beta_0, \beta_1) = \left\{ \prod_{i=1}^{n} \mathrm{Bin}(y_i; \boldsymbol{\beta}) \right\} g(\beta_0, \beta_1)$$

- We can **condition on the data** – i.e., compute the conditional distribution:

$$\pi(\beta_0, \beta_1 | y_1, y_2, \ldots, y_n) = \frac{f(y_1, y_2, \ldots, y_n, \beta_0, \beta_1)}{f(y_1, y_2, \ldots, y_n)}$$

(because everything is a r.v.)

To be continued.......

4. How to do Bayesian Analysis: MCMC

The posterior:

$$\pi(\beta_0, \beta_1 | y_1, y_2, \ldots, y_n) = \frac{f(y_1, y_2, \ldots, y_n, \beta_0, \beta_1)}{f(y_1, y_2, \ldots, y_n)}$$

• Computing the denominator is computationally expensive, and sometimes not even possible.

• Usually not recognizable as a “named” distribution.

• So we use Markov chain Monte Carlo, **MCMC**: simulation methods for *sampling* from the posterior distribution which do not require that we know the denominator, or ever have to evaluate it.

4. How to do Bayesian Analysis: MCMC

We calculate features of the posterior distribution from the posterior samples using *Monte Carlo averages*.

e.g., if we obtain $p^{(1)}, p^{(2)}, \ldots, p^{(M)}$ from the posterior distribution then

$$\tilde{\mathrm{E}}(p|y) = \frac{1}{M} \sum_i p^{(i)}$$

Monte Carlo error exists: Difference between MC average and the feature it's meant to estimate. Controllable by increasing M.

The topic of MCMC is too vast to cover here.

We use a "Metropolis within Gibbs sampling" algorithm for everything, or let BUGS deal with it.

Metropolis-within-Gibbs Sampling

Iterative sampling of parameters from the *full conditional* distributions (one for **each** parameter):

$$[\beta_0|\mathbf{y}, \beta_1] = [\mathbf{y}|\beta_0, \beta_1][\beta_0]/[\mathbf{y}]$$

$$[\beta_1|\mathbf{y}, \beta_0] = [\mathbf{y}|\beta_0, \beta_1][\beta_1]/[\mathbf{y}]$$

In many/most problems we cannot compute $[\mathbf{y}]$ and therefore cannot identify the fc as a "named" distribution to simulate from. So rules are invented for drawing random variables from distributions that cannot be identified precisely.

The Metropolis algorithm is one such rule.

The Metropolis Algorithm

(1) For a candidate value β_0^* simulated from some symmetric proposal distribution: Symmetric if $h(x|y) = h(y|x)$ (**not** part of the model)

(2) Accept that value with probability

$$r = [\beta_0^*|\mathbf{y}, \beta_1]/[\beta_0|\mathbf{y}, \beta_1]$$

(3) Do also for β_1

Handy stuff:

- The marginal distribution of $\mathbf{y}$ (i.e., denominator of the fc) cancels, so we don't need to know what it is.
- To use the Metropolis algorithm we only have to *evaluate* known distributions

A Basic MCMC Algorithm: Logistic regression

(0) Pick starting values for β_0 and β_1. At iteration 1 of the algorithm, these are the *current* values of the parameters

(1) Generate a *candidate* value of β_0 from a symmetric proposal distribution. e.g., centered on the current value:

$$\beta_0^* \sim \text{Normal}(\beta_0, \delta^2)$$

(**The proposal is not part of the model!**)

(2) Accept the candidate value with probability:

$$r = [\beta_0^* | \mathbf{y}, \beta_1]/[\beta_0 | \mathbf{y}, \beta_1]$$

then β_0^* becomes the current value.

(3) Repeat for each parameter in the model.

Remarks on Metropolis-within-Gibbs Algorithm

- Heuristic: This algorithm has us simulate candidate values somehow and then accept values that have higher posterior probability (conditional on other parameters).

- The long-run frequency of “accepted” values is that of the target posterior density!

- Note: If the prior is constant, this MCMC calculation is based on repeated evaluations of the likelihood only. So, if you write a function to do MLE you can also do MCMC.

- We have a function to evaluate the likelihood for a given value of the parameters. Using MCMC we have to do this over-and-over again....

MCMC (MwiG) for Logistic Regression

1. What prior should we use?

$$
\begin{aligned}
\beta_0 &\sim Norm(0, 10) \\
\beta_1 &\sim Norm(0, 10)
\end{aligned}
$$

2. The full conditionals look like:

$$[\beta_0|\mathbf{y}, \beta_1] \propto \left\{\prod_i Bin(y_i|\beta_0, \beta_1)\right\} g(\beta_0)$$

3. In **R**, this looks like:

```
beta0<- some value
beta1<- some value
fc.beta0<- exp(-1*negLogLike(c(beta0,beta1),y,x))*dnorm(beta0,0,10)
fc.beta1<- exp(-1*negLogLike(c(beta0,beta1),y,x))*dnorm(beta1,0,10)
```

Implementation for simulated data:

```
niter <- 50000
out <- matrix(NA, niter, 2)  # create a matrix to save the MCMC output
colnames(out) <- c("beta0", "beta1")

# Initialize the parameters, likelihood, and priors
beta0 <- starting.values[1]
beta1<-  starting.values[2]
loglike <- -1*negLogLike(c(beta0,beta1), z, vegHt)
logprior <- dnorm(c(beta0,beta1), 0, 10, log=TRUE)

for(i in 1:niter) {
    # propose candidate values of beta
    beta0.cand <- rnorm(1, beta0, 0.3) # 0.3 is tuning parameter

    # evaluate likelihood and priors for candidates
    loglike.cand <- -1*negLogLike(c(beta0.cand,beta1), z, vegHt)
    logprior.cand <- dnorm(beta0.cand, 0, 10, log=TRUE)

    # Compute Metropolis acceptance probability
    Metrop.acceptance.prob <- exp((loglike.cand+logprior.cand) -
                             (loglike + logprior[1]))

    # Keep the candidates if they meet the criterion
    if(runif(1) < Metrop.acceptance.prob) {
        beta0 <- beta0.cand
        loglike <- loglike.cand
        logprior[1] <- logprior.cand
    }
 #### Repeat for beta1
.
.
.
}  # closes main MCMC loop
```

5. Inference for Hierarchical Models

HMs have 1 or more “intermediate” models/levels/stages involving a latent variable (random effect).

Two canonical examples of HMs in ecology:

- Modeling species occurrence – “occupancy models”
- Modeling species abundance – “N-mixture models” (and related)

Example: Modeling occurrence of a species

- $y_i = \{0, 1\}$ observations of presence/absence at site i
- $z_i = \{0, 1\}$ state-variable true presence or absence

Observation model:

$$y_i | z_i \sim \text{Bernoulli}(z_i p)$$

$p =$ probability of detecting species *given that it is present*

Process model:

$$z_i \sim \text{Bernoulli}(\psi_i)$$

$$logit(\psi_i) = \beta_0 + \beta_1 x_i$$

Example: Modeling Abundance from Point Counts

- y_i = count of birds at point i
- N_i = state-variable population size at point i

Observation model:

$$y_i | N_i \sim \text{Binomial}(N_i, p)$$

N_i = local population size

p = probability of encountering an *individual*

Process model:

$$N_i \sim \text{Poisson}(\lambda_i)$$

$$log(\lambda_i) = \beta_0 + \beta_1 x_i$$

Analysis of HMs

- In addition to “data” and “parameters” hierarchical models have one or more intermediate levels of latent variables or random effects.

- Bayesian analysis proceeds without any novel considerations. We apply MCMC to the joint distribution and we also update the random effects using our MCMC algorithm.

- Classical analysis by likelihood. We have to compute the marginal likelihood by removing the latent variables by integration or summation.

Bayesian Analysis

- Nothing is special about this model. Since everything is a random variable we can compute the joint distribution:

$$f(y_1, y_2, \ldots, y_n, z_1, \ldots, z_n, p, \boldsymbol{\beta}) = \left\{ \prod_{i=1}^{n} \prod_{j=1}^{J} [y_{ij}|z_i, p][z_i|\boldsymbol{\beta}] \right\} [\boldsymbol{\beta}]$$

- Posterior characterized by MCMC – Metropolis/Gibbs:

parameters:

$$[\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{z}]$$
$$[p|\mathbf{y}, \boldsymbol{\beta}]$$

latent variables:

$$[z_i|\mathbf{y}, \boldsymbol{z}_{-i}, p, \boldsymbol{\beta}] = [z_i|\mathbf{y}, p, \boldsymbol{\beta}]$$

The posterior is fully characterized by this set of conditional distributions

Classical Analysis of Random Effects

INTEGRATED (marginal) LIKELIHOOD:

- Remove random effects from the conditional likelihood by integration

- The distribution of $\mathbf{y}_i$ *unconditional* on the random effect:

$$f(\mathbf{y}_i|\boldsymbol{\beta}, p) = \int \left\{ \prod_{j=1}^{J} f(y_{ij}|z_i, p) \right\} g(z_i|\boldsymbol{\beta}) dz_i$$

- Not a function of z_i anymore

- Maximize to obtain MLEs of p, $\boldsymbol{\beta}$

- For discrete latent variable, replace $\int$ by Σ

Example: Occupancy Model

Observation model:

$$\begin{aligned} y_{ij} &\sim Bin(J,p) \quad \text{if } z_i = 1 \\ y_{ij} &= 0 \qquad\qquad \text{if } z_i = 0 \end{aligned}$$

State model:

$$z_i \sim Bin(\psi_i)$$

$$logit(\psi_i) = \beta_0 + \beta_1 x_i$$

What is the marginal likelihood for y?

Computing the Marginal Likelihood

- z is a discrete random variable
- Law of total probability

$$Pr(y) = Pr(y|z=1)Pr(z=1) + Pr(y|z=0)Pr(z=0)$$

- Marginal likelihood for total detections, y_i:

$$[y_i|p,\psi] = Bin(y_i|p)\psi + 1(y_i = 0)(1-\psi)$$

- Zero-inflated binomial. Can be maximized easily to obtain MLEs.
- PRESENCE or `unmarked` function `occu`

Doing it in R

```
nSites <- 100
vegHt <- runif(nSites, 1, 3) # uniform from 1 to 3
psi <- plogis(-3 + 2*vegHt)

# Now we simulated true presence/absence for 100 sites
z <- rbinom(nSites, 1, psi)

## Now generate observations
p<- 0.6
J<- 3  # sample each site 3 times
y<-rbinom(nSites,J,p*z)

# This is the negative log-likelihood.
negLogLikeocc <- function(beta, y, x,J) {
    beta0 <- beta[1]
    beta1 <- beta[2]
    p<- plogis(beta[3])

    psi <- plogis(beta0 + beta1*x)

    marg.likelihood <- dbinom(y, J,p)*psi + ifelse(y==0,1,0)*(1-psi)
    return(-sum(log(marg.likelihood)))
}

starting.values <- c(beta0=0, beta1=0,logitp=0)
opt.out <- optim(starting.values, negLogLikeocc, y=y, x=vegHt,J=J,
                 hessian=TRUE)
```

Summary Thoughts on Bayes vs non-Bayes

Both inference paradigms useful for analysis of hierarchical models

- Bayesian:
  - Completely general methods for implementation (MCMC) which always work. Sometimes BUGS implementations don't work, so its good to know how to do it.
  - Takes more math/programming know-how????
  - Sometimes slower due to more calculations
  - Inferences are *not* asymptotic, apply to arbitrary n
  - Prediction is more coherent – comes "for free"
- Classical:
  - Integrated likelihood sometimes not feasible. (community model)
  - But very accurate (no MC error)
  - Automatic model selection (AIC)

“Bayesian” Hierarchical Model??

- Hierarchical modeling is a conceptual and technical framework for formulating models

- The method of inference is independent of model formulation

- HMs can be analyzed by Bayesian and non-Bayesian methods.

- HM $\neq$ Bayesian!!! A model is not Bayesian or frequentist – what you do to that model is Bayesian or frequentist.

- Models don’t have political views (people do)